# Data Augmentation for Deep Neural Network Acoustic Modeling

Xiaodong Cui, *Senior Member, IEEE*, Vaibhava Goel, *Senior Member, IEEE*, and Brian Kingsbury, *Senior Member, IEEE*

***Abstract*—This paper investigates data augmentation for deep neural network acoustic modeling based on label-preserving transformations to deal with data sparsity. Two data augmentation approaches, vocal tract length perturbation (VTLP) and stochastic feature mapping (SFM), are investigated for both deep neural networks (DNNs) and convolutional neural networks (CNNs). The approaches are focused on increasing speaker and speech variations of the limited training data such that the acoustic models trained with the augmented data are more robust to such variations. In addition, a two-stage data augmentation scheme based on a stacked architecture is proposed to combine VTLP and SFM as complementary approaches. Experiments are conducted on Assamese and Haitian Creole, two development languages of the IARPA Babel program, and improved performance on automatic speech recognition (ASR) and keyword search (KWS) is reported.**

***Index Terms*—Data augmentation, stochastic feature mapping, deep neural networks, automatic speech recognition, keyword search.**

## I. Introduction

ACOUSTIC modeling based on deep neural networks (DNNs) has established the state-of-the-art performance for automatic speech recognition (ASR) in the last few years [1]–[6]. However, performance degradation is often observed when the training data is sparse. For instance, in the IARPA Babel program [7], the ASR performance drops dramatically when the amount of training data is reduced from 100 hours to 10 hours [8]–[10]. This paper aims at improving the performance of deep neural network acoustic models under data sparsity by data augmentation based on label-preserving transformations.

In pattern recognition problems, an artificial neural network can be learned as a classifier using training samples with their labels. Patterns belonging to the same class under the same label have variations. If there are sufficient patterns from each class for training, the neural network will learn from the abundant variations presented in the training samples under the same labels and make classifications that are invariant to such variations. When the training samples are not sufficient, the pattern variations are limited due to data sparsity and the trained neural network classifier will have poor classification invariance. Under this condition, data augmentation based on label-preserving transformations can help to alleviate the sparse data issue. Label-preserving transformations artificially generate more training samples by transforming the existing training samples using certain forms of transformations that preserve the class labels. The idea behind label-preserving transformations is to increase the pattern variations through the transformations to improve the classification invariance and generalization ability of the neural networks.

Data augmentation using label-preserving transformations has been widely used in neural network based pattern recognition in computer vision and image recognition tasks [11]–[13] where transformations such as translation, deformation and reflection have led to significant improvements in recognition accuracy. In speech recognition, similar approaches have also been put into practice in Gaussian mixture model(GMM)/hidden Markov model(HMM) based acoustic modeling, although sometimes under different terminology. Multi-style training (MST)[14]–[16], which is commonly used for noise robust speech recognition by artificially adding noisy data with various types and levels of noise to the original training samples, can be viewed as a data augmentation strategy making use of label-preserving transformations. Another example is IMELDA [17] where multi-condition transformations are learned from tilted, noisy and undegraded speech data so that the sensitivity of the transformations to those conditions is reduced.

Recently, efforts on using data augmentation for low resource ASR with deep neural networks have been reported. A strategy based on vocal tract length perturbation (VTLP) was proposed in [18] and experiments on the TIMIT database using deep convolutional neural networks (CNNs) showed decent improvements in phone error rate (PER). The work was later followed up by [19]–[21] on large vocabulary continuous speech recognition (LVCSR). Similarly, elastic spectral distortion was investigated in [22] where sparse data was augmented by vocal tract length (VTL) distortion, speech rate distortion and frequency-axis random distortion for DNN-HMM training.

In terms of pattern variations, speech signals can be affected by a wide variety of factors such as speaker, gender, age, accent, channel and environment. Some of the speech variabili-

Manuscript received March 09, 2015; accepted May 08, 2015. Date of publication June 01, 2015; date of current version June 04, 2015. This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) under contract number W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Vincent Vanhoucke

The authors are with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: cuix@us.ibm.com; vgoel@us.ibm.com; bedk@us.ibm.com).

Digital Object Identifier 10.1109/TASLP.2015.2438544

ties are not straightforward to generate via simple transformations. In general, compared to image recognition, data augmentation based on label-preserving transformations is less known in speech recognition, especially for DNN-HMMs. So far most of the reported approaches in ASR rely on spectral alteration by perturbing or distorting the original speech spectra.

In this paper, we exploit data augmentation approaches to dealing with limited training data in DNN/CNN acoustic modeling for LVCSR. Other than investigating VTLP on LVCSR as a representative data augmentation approach, we propose a novel label-preserving transformation method by the name of stochastic feature mapping (SFM). SFM performs statistical voice conversion between speakers in a designated feature space, which does not rely on spectral alteration. We will demonstrate how VTLP and SFM can be applied to both DNNs with a speaker-adaptive input feature space and CNNs with a LogMel input feature space, respectively. Furthermore, we also propose a two-stage data augmentation scheme based on a stacked architecture that combines VTLP and SFM as two complementary approaches.

Experiments are carried out in the context of IARPA Babel program [7]. The program intends to develop agile and robust ASR and keyword search (KWS) technologies that can be rapidly applied to any human language to enable analysts to efficiently process massive amounts of real-world recorded speech. The program aims at fostering innovations in ASR and KWS for a much larger set of languages than has hitherto been addressed, with significantly less training data. The data augmentation approaches investigated in this paper are evaluated on Assamese and Haitian Creole, two development languages of the Babel option period one.

The remainder of the paper is organized as follows. Section II is devoted to the details of VTLP and SFM. Section III shows how VLTP and SFM can be used for training DNN and CNN acoustic models. It also introduces the two-stage data augmentation scheme that combines VTLP and SFM in a stacked architecture. Experimental results on ASR and KWS on Assamese and Haitian Creole are presented in Section IV followed by a discussion in Section V. Finally we conclude the paper with a summary in Section VI.

## II. Data Augmentation Approaches

### A. Vocal Tract Length Perturbation (VTLP)

VTLP, which was first proposed in [18], is representative of a group of data augmentation schemes that generate new samples through perturbing or distorting the speech spectra of the existing training samples. In [18], for each utterance in the training set, a warping factor $\alpha$ is randomly chosen from [0.9, 1.1] to warp the frequency axis. Therefore, the VTL of the speaker is perturbed and a new replica of the utterance is created under the distortion of the original spectrum of the utterance.

In this paper a modified version of VTLP is adopted in which the VTL warping factor $\alpha$ of a speaker is first estimated and then perturbed deterministically in both positive and negative directions as follows:

$$\alpha \mapsto \{\alpha \pm \Delta, \cdots, \alpha \pm k\Delta, \cdots, \alpha \pm K\Delta\} \quad k = 1, \cdots, K \tag{1}$$

where $2K$ is the total number of replicas of the original data and $\Delta$ is a fixed (positive) shift along the $\alpha$ axis. The perturbed VTL warping factors are then used to re-scale the frequency axis of the original speech spectra to create new replicas.

The reason behind choosing the deterministic perturbation over the random perturbation in [18] is that speech features such as Mel-frequency cepstral coefficients (MFCC) or perceptual linear prediction (PLP) coefficients are not very sensitive to a small distortion of the VTL warping factor. When random perturbation is used, especially when the number of replicas is relatively large, there is a chance that two perturbed VTL warping factors are close to each other. To guarantee an effective perturbation, we force a fixed minimum gap ($\Delta$) between perturbed warping factors. Based on our empirical observation on pilot experiments using the IBM Babel systems, the deterministic perturbation strategy gives a slightly superior performance over the random perturbation. In practice, VTLP with both random [20] and deterministic [21] perturbation has been used in different systems and both have improved performance.

### B. Stochastic Feature Mapping (SFM)

Inspired by the idea of voice conversion [23]–[25], SFM augments training samples by statistically converting one speaker's speech data to another speaker's. Mathematically, it seeks to address the following problem.

Given a feature space $\boldsymbol{\mathcal{H}}$, suppose there is a speaker $S$ who speaks an utterance $\mathbf{u}$ with label $\mathbf{W}$ which generates a sequence of features with $N$ frames

$$\boldsymbol{\mathcal{O}}^{(\mathsf{S})} = \{\mathbf{o}_1^{(\mathsf{S})}, \cdots, \mathbf{o}_N^{(\mathsf{S})}\}, \mathbf{o}_t^{(\mathsf{S})} \in \boldsymbol{\mathcal{H}} \tag{2}$$

Then for another speaker, $T$, what would the sequence of features

$$\boldsymbol{\mathcal{O}}^{(\mathsf{T})} = \{\mathbf{o}_1^{(\mathsf{T})}, \cdots, \mathbf{o}_N^{(\mathsf{T})}\}, \mathbf{o}_t^{(\mathsf{T})} \in \boldsymbol{\mathcal{H}} \tag{3}$$

be if he/she were to speak the same utterance $\mathbf{u}$ under the same label $\mathbf{W}$? In what follows we refer to the speaker $S$ as the source speaker and the speaker $T$ as the target speaker.

To create a mapping from the source speaker to the target speaker, SFM first obtains a speaker dependent acoustic model $\boldsymbol{\lambda}_{\boldsymbol{\mathcal{H}}}^{(\mathsf{T})}$ of the target speaker from $\boldsymbol{\mathcal{O}}^{(\mathsf{T})}$ in the feature space $\boldsymbol{\mathcal{H}}$ and then estimates a transformation $\boldsymbol{\mathcal{F}}(\cdot)$ on $\boldsymbol{\mathcal{O}}^{(\mathsf{S})}$ such that it minimizes a chosen objective function $\boldsymbol{\mathcal{L}}$:

$$\hat{\boldsymbol{\mathcal{F}}} = \arg\min_{\boldsymbol{\mathcal{F}}} \boldsymbol{\mathcal{L}}\left(\boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{O}}^{(\mathsf{S})}), \boldsymbol{\lambda}_{\boldsymbol{\mathcal{H}}}^{(\mathsf{T})}\right) \tag{4}$$

Once the transformation $\boldsymbol{\mathcal{F}}$ is estimated, which is obviously label preserving, it is used to map all the data from the source

speaker to the target speaker. Algorithm I summarizes the generic SFM algorithm.

**Algorithm 1** Data Augmentation by Stochastic Feature Mapping

K $\leftarrow$ number of replicas;
M $\leftarrow$ number of speakers;
**for** $i \leftarrow 1, \cdots, M$ **do**
  Estimate speaker dependent model $\boldsymbol{\lambda}_{\boldsymbol{\mathcal{H}}}^{(i)}$ in feature space $\boldsymbol{\mathcal{H}}$ using all the data available from speaker $i$;
**end for**
**for** $i \leftarrow 1, \cdots, M$ **do**
  **for** $k \leftarrow 1, \cdots, K$ **do**
    Randomly select a new speaker $k$ as the target speaker;
    Estimate transformation $\boldsymbol{\mathcal{F}}$ based on model $\boldsymbol{\lambda}_{\boldsymbol{\mathcal{H}}}^{(k)}$ and all utterances from speaker $i$ in feature space $\boldsymbol{\mathcal{H}}$ minimizing the objective function $\boldsymbol{\mathcal{L}}(\boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{O}}^{(i)}), \boldsymbol{\lambda}_{\boldsymbol{\mathcal{H}}}^{(k)})$;
    Map all utterances from speaker $i$ to the target speaker $k$ using $\boldsymbol{\mathcal{F}}$ in feature space $\boldsymbol{\mathcal{H}}$;
  **End for**
**End for**

SFM performs "voice conversion" statistically in the sense of stochastic mapping, which does not rely on deterministic spectral alteration. Furthermore, distinct from the conventional voice conversion techniques [23]–[25] which are usually carried out in the spectral space, SFM can convert "voice" in any designated feature space for the convenience of acoustic model training. The speaker dependent model of the target speaker is expected to reflect the acoustic characteristics of the speaker including those speech variabilities (e.g. age and accent) that are not obvious to map directly using simple transformations. Therefore, based on the speaker dependent model, SFM can implicitly transform such speech variabilities in the augmented training data.

## III. Data Augmentation for DNN/CNN Models

Data augmentation algorithms are closely related to the input feature spaces of neural networks. In this section, we will show how VTLP and SFM can be applied to deep neural network acoustic modeling with some commonly-used input features. Specifically, we will investigate VTLP and SFM for DNN models with speaker adaptive input features and for CNN models with LogMel input features, respectively. Extension to other features should be straightforward.

### A. Deep Neural Networks

The IBM DNN systems in the Babel program [26] use a speaker adaptive input feature space shown in Fig. 1. In this feature extraction pipeline 13-dimensional PLP features are used as the preliminary acoustic features after cepstral mean normalization (CMN) and vocal tract length normalization (VTLN)[27]. After taking into account the context (CTX) information by splicing 9 adjacent frames, linear discriminant analysis (LDA) is

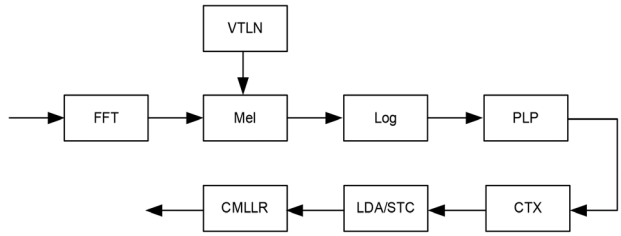



Fig. 1. Speaker adaptive feature processing for DNN model input.

used to project the feature dimensionality down to 40. The components of LDA features are further decorrelated by a global semi-tied covariance (STC) matrix [28]. In this LDA space, speaker adaptive training (SAT) using constrained maximum likelihood linear regression (CMLLR) [29] is further applied. Finally, the DNN input layer takes 9 consecutive frames of such speaker-adapted features as its input.

For VTLP, the implementation is trivial. After the VTL is estimated, perturbation is performed according to Eq. (1) before the VTLN step.

For SFM, the selected space for feature mapping is the LDA space. First of all, a speaker dependent model is built for the target speaker $T$ in the LDA space $\boldsymbol{\lambda}_{\mathsf{LDA}}^{(\mathsf{T})}$, which is accomplished by model space maximum likelihood linear regression (MLLR) [30] based on a regression tree. Given $\boldsymbol{\lambda}_{\mathsf{LDA}}^{(\mathsf{T})}$, assume the transformation function $\boldsymbol{\mathcal{F}}$ has a linear form

$$\boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{O}}) = \mathbf{A}\boldsymbol{\mathcal{O}} + \mathbf{b} \tag{5}$$

and estimate it under the maximum likelihood (ML) criterion:

$$\{\tilde{\mathbf{A}}, \tilde{\mathbf{b}}\} = \arg \max_{\{\mathbf{A},\mathbf{b}\}} \log P(\mathbf{A}\boldsymbol{\mathcal{O}}_{\mathsf{LDA}}^{(\mathsf{S})} + \mathbf{b} | \boldsymbol{\lambda}_{\mathsf{LDA}}^{(\mathsf{T})}) \tag{6}$$

which is equivalent to a CMLLR setup. Using the estimated linear transformation $\boldsymbol{\mathcal{F}}$, the feature sequence of the source speaker can be transformed into that of the target speaker in the LDA space

$$\boldsymbol{\mathcal{O}}_{\mathsf{LDA}}^{(\mathsf{T})} = \tilde{\mathbf{A}}\boldsymbol{\mathcal{O}}_{\mathsf{LDA}}^{(\mathsf{S})} + \tilde{\mathbf{b}} \tag{7}$$

Going back the feature extraction pipeline in Fig. 1, assuming $\{\mathbf{A}^{(\mathsf{T})}, \mathbf{b}^{(\mathsf{T})}\}$ is the CMLLR transformation after the LDA space for the target speaker $T$, then one has the final speaker adaptive features as

$$\begin{aligned} \boldsymbol{\mathcal{O}}_{\mathrm{CMLLR}}^{(\mathsf{T})} &= \mathbf{A}^{(\mathsf{T})}\boldsymbol{\mathcal{O}}_{\mathsf{LDA}}^{(\mathsf{T})} + \mathbf{b}^{(\mathsf{T})} \\ &= \mathbf{A}^{(\mathsf{T})}(\tilde{\mathbf{A}}\boldsymbol{\mathcal{O}}_{\mathsf{LDA}}^{(\mathsf{S})} + \tilde{\mathbf{b}}) + \mathbf{b}^{(\mathsf{T})} \end{aligned} \tag{8}$$

By inspecting Eq. (8) one can see that in this case SFM is carried out via a composition of two linear transformations: one maps a feature sequence of the source speaker to that of the target speaker in the LDA space and the other transforms the mapped feature sequence from the LDA space to the speaker adaptive space using CMLLR specific to the target speaker. Attention should to be paid to Eq. (8). Since the two linear functions in the composition are both estimated under ML, they should not be estimated jointly at the same time. Otherwise, the latter linear function will absorb the former and the composition of two is meaningless for the feature mapping.

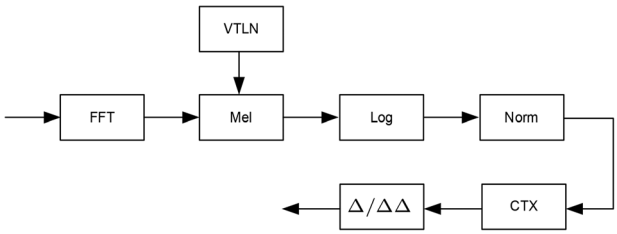



Fig. 2. LogMel feature processing for CNN model input.

### B. Convolutional Neural Networks

LogMel features, which are topographical in representation, are often used as input features to CNN models. Fig. 2 shows a mean and speaker normalized LogMel feature extraction pipeline used in IBM CNN systems [31][32]. This pipeline computes VTLN logMel features and subtracts the speaker-dependent mean. The normalized logMel features are spliced with their left and right 5 adjacent frames to form a feature map. Two other feature maps are created by computing deltas and double deltas.

Given the normalized LogMel input features, the extension of VTLP to CNN is straightforward. Eq. (1) can be directly applied after VTLs are estimated.

For SFM, we still use the same strategy in Section III-A [33]. The selected feature space for feature mapping is the LogMel space, in which a speaker-dependent model is first estimated for the target speaker using model space MLLR. A linear mapping is chosen as the transformation function $\boldsymbol{\mathcal{F}}$ and it is to be estimated under the ML criterion. Again, it is equivalent to CMLLR. However, different from the speaker-adaptive processing discussed in Section III-A where the LDA features are decorrelated after STC, the dimensions of the LogMel features, which are the outputs of Mel-frequency filter bank, are strongly correlated. This poses a problem for the conventional CMLLR estimation which assumes diagonal covariances in GMMs [29]. One way to deal with this problem is to diagonalize the LogMel features and estimate the CMLLR transformation in the diagonalized space, after which the features are transformed back to the original LogMel space. In this work, the diagonalization is accomplished by a global STC transformation. The mapping from the source speaker $S$ to the target speaker $T$ is indicated in Eq. (9):

$$\mathbf{O}_{\mathsf{LogMel}}^{(\mathsf{T})} = \mathbf{C}^{-1} \cdot \boldsymbol{\mathcal{F}} \cdot \mathbf{C} \cdot \mathbf{O}_{\mathsf{LogMel}}^{(\mathsf{S})} \tag{9}$$

where $\mathbf{C}$ is the global STC transformation and $\mathbf{C}^{-1}$ is its inverse. $\boldsymbol{\mathcal{F}}$ is the (augmented) CMLLR transformation in the diagonalized LogMel feature space. Note that the speaker-dependent model of the target speaker should also be estimated in the diagonalized feature space after STC when used for estimating the CMLLR transformation.

The joint use of STC and CMLLR has been shown effective for transforming correlated features. It has been previously used in speaker adaptation for DNN and CNN acoustic models with filter bank input features in LVCSR tasks [31][34].

### C. A Two-stage Data Augmentation Scheme

VTLP and SFM generate speech variations from different perspectives. By perturbing the VTL of a speaker, VLTP creates new VTLs and hence "new" speakers. On the other hand, SFM does not create new speakers. Instead, it increases the utterances from each speaker in the training set by cross-mapping voices between the existing speakers. Therefore, it improves the acoustic richness in the training data. Based on such observations, there is a reason to believe that the two approaches can be complementary. In this section, we investigate a two-stage data augmentation scheme, as illustrated in Fig. 3, based on a stacked architecture that combines the merits of VTLP and SFM [33].

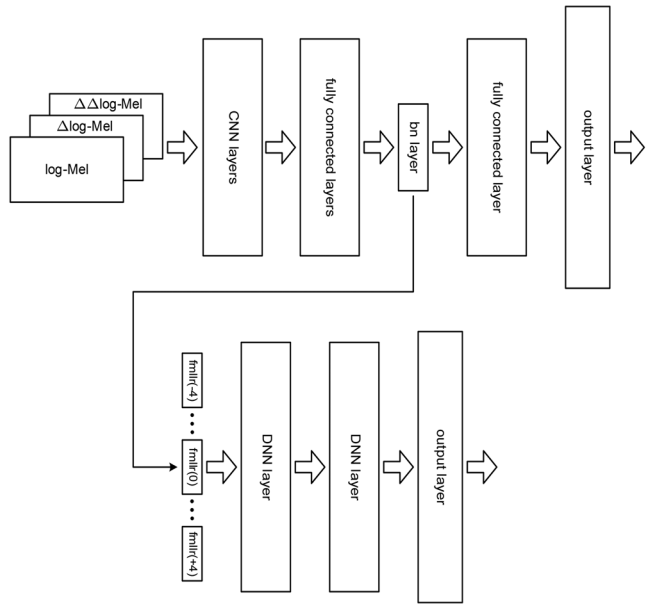



Fig. 3. A stacked architecture that integrates VTLP and SFM as complementary approaches in a two-stage data augmentation scheme.

The two-stage scheme increases speech variations one step at a time. In the first stage, a bottleneck CNN is built with LogMel input discussed in Section III-B. The training data for the CNN is augmented by VTLP. The bottleneck CNN is used as a feature extractor where the input to the sigmoid nonlinear activation function of the bottleneck layer is employed as features for the next stage. Since the CNN is trained with VTLP, the features extracted this way are expected to be more speaker invariant than the original features. In the second stage, a DNN is built whose input is speaker-adapted bottleneck features extracted from the CNN trained in the first stage. The training data for the DNN is augmented by SFM to further improve its acoustic richness. The SFM conducted in this stage is analogous to that discussed in Section III-A. The feature space in which the speaker dependent acoustic model of the target speaker and the mapping between the source and target speakers are estimated is the bottleneck feature space. Details of the CNN and DNN configurations will be described in the experimental section.

## IV. Experiments

ASR and KWS experiments are carried out on Assamese and Haitian Creole, two development languages from the option period one of the IARPA Babel program. For each language, there are two language packs: full language pack (FLP) and limited language pack (LLP). The LLPs are a subset of the FLPs. The Assamese comprises 154.0 hours of data in the FLP training set and 24.3 hours of data in the LLP training set.

Its development set comprises 20 hours of data. The Haitian Creole comprises 138.8 hours of data in the FLP training set and 23.8 hours of data in the LLP training set. Its development set also comprises 20 hours of data. Both training data sets consist of conversational and scripted speech while the development sets consist of conversational speech only. Specifically, the Assamese FLP is composed of 119.7 hours of conversational data (790 speakers) and 34.3 hours of scripted data (344 speakers). Its LLP is composed of 20.0 hours of conversational data (138 speakers) and 4.3 hours of scripted data (44 speakers). The Haitian Creole FLP is composed of 120.5 hours of conversational data (760 speaker) and 18.3 hours of scripted data (315 speakers). Its LLP is composed of 19.9 hours of conversational data (126 speakers) and 3.9 hours of scripted data (47 speakers). All the data is telephony, sampled at 8 KHz. Approximately 50% of the audio is speech. This effort uses the IARPA Babel program language collection releases IARPA-babel102b-v0.5a and IARPA-babel201b-v0.2b full and limited language packs.

Based on the Babel option period one evaluation results of five development languages (Assamese, Bengali, Haitian Creole, Zulu and Lao), Assamese is one of the harder languages with performance on the low end while Haitian Creole is one of the easier languages with performance on the high end. The selection of these two languages in this paper attempts to be representative in terms of performance. While experimental results on FLPs will be covered, the main focus of the paper is on the impact of data augmentation on LLPs, the limited training data case. In what follows, ASR and KWS experimental results will be reported, respectively.

### *A. ASR Experiments*

The baseline DNN acoustic model has 5 hidden layers of 1024 hidden units with sigmoid activation functions and a softmax output layer. The input to the network is 9 adjacent frames of 40-dimensional speaker adaptive features described in Section III-A. In the FLP configuration, the output layer has 3,000 units derived from the context-dependent states of a GMM-HMM acoustic model. In the LLP configuration, the output layer has 2,000 units derived from the context-dependent states of a GMM-HMM acoustic model after bootstrap and restructuring [26][35].

The baseline CNN model has two convolutional layers followed by five fully connected feedforward layers. All hidden layers use sigmoid activation functions and the output layer is softmax with 2,000 units for the LLP configuration. The input features to the first convolutional layer are three feature maps including 40-dimensional LogMel features, their deltas and double deltas described in Section III-B. The temporal context is 11 frames. There are 128 hidden units (feature maps) in the first convolutional layer, the local receptive field has an overlapping window of $9 \times 9$ with a shift of 1 in both temporal and spectral domains, which results in $32 \times 3$ windows for each feature map. On top of that, max pooling is applied in a $3 \times 1$ non-overlapping window which results in $11 \times 3$ windows for each feature map. There are 256 hidden units (feature maps) in the second convolutional layer, the local receptive field has an overlapping window of $4 \times 3$ with a shift of 1 in both temporal and spectral domains which results in $8 \times 1$ windows for each feature map. Following the second convolutional layer are five fully connected feedforward layers, each containing 1,024 hidden units.

For the two-stage data augmentation scheme, the bottleneck CNN in the first stage has two convolutional layers followed by six fully connected feedforward layers among which the penultimate layer is a bottleneck layer with 40 dimensions. All fully connected layers including the bottleneck layer use sigmoid activation functions. Other than the bottleneck layer, the CNN has the same configuration as the baseline CNN. The input to the sigmoid nonlinear function of the bottleneck layer is used as the features for the next stage. The reason to use the input instead of the output of the sigmoid nonlinearity is to ensure a good dynamic range. Furthermore, the resulting linear output is more suitable for Gaussian assumption, which will benefit the GMM based speaker adaptive training in the next step. The DNN in the second stage has two hidden layers and each layer has 1024 hidden units with sigmoid activation functions.

Both DNNs and CNNs are initialized with layer-wise discriminative pre-training. After the pre-training, the networks are first fine-tuned by 15 iterations of cross-entropy (CE) training and then followed by 30 iterations of Hessian-free (HF) sequence training based on the state-level minimum Bayes risk (sMBR) criterion [4].

For data augmentation, both VTLP and SFM generate 4 replicas of the original data and only the conversational data in the training set is augmented. The VTLP is implemented using the IBM Attila toolkit [36] where VTL warping factors are quantized between $[0.8, 1.25]$. As a result, the estimated warping factor $\alpha$ is an integer between $[0, 20]$ with 10 equivalent to the neutral warping factor 1.0. The perturbed warping factors, if they are beyond $[0.8, 1.25]$, are clipped to 0.8 or 1.25 which corresponds to integer 0 or 20, respectively. Specifically, $K = 2$ and $\Delta = 2$ in Eq. (1) in this implementation. In the two-stage data augmentation scheme, 4 replicas created by VTLP are used for the training of the first-stage CNN bottleneck models and additional 4 replicas created by SFM (8 replicas in total) are used for the training of the second-stage DNN models.

*1) VTLP vs. SFM:* The performance of VTLP and SFM on Assamese and Haitian Creole LLPs using DNNs and CNNs is demonstrated in Table I. The baseline models are trained without data augmentation. For both DNN and CNN models, WER reductions are observed when using VTLP and SFM. Specifically, for DNN models, VTLP yields 1.6% and 3.2% absolute improvements on Assamese and Haitian Creole, respectively; SFM yields 2.2% and 3.7% absolute improvements on Assamese and Haitian Creole, respectively. Similarly, for CNN models, VTLP yields 3.7% and 2.8% absolute improvements on Assamese and Haitian Creole, respectively; SFM yields 3.2% and 2.5% absolute improvements on Assamese and Haitian Creole, respectively. SFM obtains larger gains than VTLP for DNN models while VTLP slightly outperforms SFM for CNN models.

When using the two-stage data augmentation scheme that combines VTLP and SFM based on the stacked architecture, it yields superior performance over both DNN and CNN models. For Assamese, the two-stage scheme achieves 0.6% absolute

TABLE I
WORD ERROR RATES (WERS) OF DNN AND CNN ACOUSTIC MODELS ON ASSAMESE AND HAITIAN CREOLE LLPS WITH AND WITHOUT DATA AUGMENTATION

| Language | Assamese | | Haitian Creole | |
|---|---|---|---|---|
| Model | DNN | CNN | DNN | CNN |
| Baseline | 66.3 | 67.4 | 62.8 | 61.2 |
| VTLPx4 | 64.7 | 63.7 | 59.6 | 58.4 |
| SFMx4 | 64.1 | 64.2 | 59.1 | 58.7 |
| Two-stage (VTLPx4+SFMx4) | 63.1 | | 57.1 | |

TABLE II
WORD ERROR RATES (WERS) OF DNN ACOUSTIC MODELS ON ASSAMESE AND HAITIAN CREOLE LLPS USING DATA AUGMENTATION WITH VARIOUS NUMBERS OF REPLICAS

| Language | Assamese | Haitian Creole |
|---|---|---|
| Baseline | 66.3 | 62.8 |
| VTLPx4 | 64.7 | 59.6 |
| VTLPx8 | 64.7 | 59.8 |
| SFMx4 | 64.1 | 59.1 |
| SFMx8 | 63.7 | 58.9 |

TABLE III
WORD ERROR RATES (WERS) OF DNN ACOUSTIC MODELS BUILT ON FLPS AND LLPS ON ASSAMESE AND HAITIAN CREOLE WITH AND WITHOUT DATA AUGMENTATION

| Language | Assamese | | Haitian Creole | |
|---|---|---|---|---|
| Model | LLP | FLP | LLP | FLP |
| Baseline | 66.3 | 52.9 | 62.8 | 48.7 |
| VTLPx4 | 64.7 | 52.5 | 59.6 | 48.0 |
| SFMx4 | 64.1 | 52.0 | 59.1 | 47.7 |

WER reduction ($63.7\% \rightarrow 63.1\%$) compared to the best performance among VTLP and SFM applied in DNN and CNN models. For Haitian Creole, it achieves 1.3% absolute WER reduction ($58.4\% \rightarrow 57.1\%$) compared the best performance of VTLP and SFM applied in DNN and CNN models.

*2) Number of Replicas:* Table II compares the performance of VTLP and SFM with various numbers of replicas for DNN models on LLPs. The baseline models are trained without data augmentation. It can be seen from the table that if we increase the replicas from 4 ($K = 2, \Delta = 2$ in Eq. (1)) to 8 ($K = 4, \Delta = 1$ in Eq. (1)), the performance of VTLP tends to plateau or even degrade slightly. That's because too small a difference in perturbation of the VTL will not make the variations created distinctive enough. On the other hand, SFM can still improve, although marginally. The increase of replicas from 4 to 8 can significantly increase the training time. Therefore, considering the tradeoff between the performance and training time, 4 replicas are used for VTLP and SFM for the rest of the experiments.

*3) LLP vs. FLP:* Table III shows the impact of data augmentation to DNN models with different amounts of training data. The baseline models are trained without data augmentation. From the table, the WER reduction by data augmentation on FLP is significantly smaller than that on LLP for both VTLP and SFM on the two languages. Specifically, the WER reduction by VTLP is 1.6% and 3.2% absolute for LLP while only 0.6 and 0.7% absolute for FLP on Assamese and Haitian Creole, respectively. Similarly, the improvement by SFM drops from 2.2% and 3.7% absolute for LLP to 0.9% and 1.0% absolute for FLP on Assamese and Haitian Creole, respectively.

The results are expected. Data augmentation is most helpful when the training data is sparse. With more training data available, there are more speech variations present in the data in the learning of the networks. Therefore, the impact of data augmentation by introducing additional pattern variants will be weakened. However, in this task, one can see that data augmentation still helps for FLPs consisting of about 150 hours of audio (approximately 70 hours of speech data) for each language.

### B. KWS Experiments

The KWS performance is measured by the term-weighted value (TWV) [37] which is defined as a function of the probability of missed detections and the probability of false alarms

$$\mathrm{TWV}(\theta) = 1 - [P_{\mathrm{miss}}(\theta) + \beta P_{\mathrm{FA}}(\theta)] \tag{10}$$

The probability of miss and the probability of false alarm are defined as follows:

$$P_{\mathrm{miss}}(\theta) = \frac{1}{K} \sum_{W=1}^{K} \frac{N_{\mathrm{miss}}(W, \theta)}{N_{\mathrm{true}}(W)} \tag{11}$$

$$P_{\mathrm{FA}}(\theta) = \frac{1}{K} \sum_{W=1}^{K} \frac{N_{\mathrm{FA}}(W, \theta)}{N_{\mathrm{NT}}(W)} \tag{12}$$

where $W$ are the keywords; $K$ is the number of keywords; $N_{\mathrm{true}}(W)$ is the number of reference occurrences of $W$; $N_{\mathrm{miss}}(W, \theta)$ is the number of missed occurrences of $W$ with threshold $\theta$; $N_{\mathrm{FA}}(W, \theta)$ is the number of incorrectly detected occurrences of $W$ at threshold $\theta$; and $N_{\mathrm{NT}}(W)$ is the number of non-target trials for $W$. Since an audio corpus for indexing is continuous in nature as oppose to discrete trials, $N_{\mathrm{NT}}(W)$ is defined as

$$N_{\mathrm{NT}}(W) = T_{\mathrm{audio}} - N_{\mathrm{true}}(W) \tag{13}$$

with $T_{\mathrm{audio}}$ the total audio duration in seconds. A rate of one trial per second is assumed in the above definition. The constant $\beta = 999.9$ sets the relative cost of false alarms vs. misses. In this paper, maximum term-weighted value (MTWV), which is the best TWV achievable by varying the threshold that defines the YES/NO decisions in the postings list, is used for measuring the KWS performance.

TABLE IV
MTWVS OF DNN AND CNN ACOUSTIC MODELS ON ASSAMESE AND HAITIAN CREOLE LLPS WITH AND WITHOUT DATA AUGMENTATION

| Language | Assamese | | Haitian Creole | |
|---|---|---|---|---|
| Model | DNN | CNN | DNN | CNN |
| Baseline | 0.2457 | 0.2257 | 0.4526 | 0.4793 |
| VTLPx4 | 0.2774 | 0.2833 | 0.4786 | 0.5070 |
| SFMx4 | 0.2826 | 0.2691 | 0.5021 | 0.4978 |
| Two-stage (VTLPx4+SFMx4) | 0.2862 | | 0.5178 | |

The IBM KWS systems are based on a two-pass implementation of weighted finite state transducer (WFST) indexing and search [26], [38], [8]. The query list is split into in-vocabulary (IV) and out-of-vocabulary (OOV) queries according to the ASR lexicon. For IV queries, each query is converted into a lexical finite state acceptor which is then composed with the lexical index. For OOV queries, each query is converted by a grapheme-to-phoneme converter to a phonetic finite-state acceptor which is then composed with the phonetic index. The composition also includes a phoneme-to-phoneme confusion model for query expansion. A cascaded strategy [39] is used in the keyword search where the word index is first searched for IV queries and if no results are returned or the query term is OOV then the phonetic index is searched. Sum-to-one normalization [40][41] is applied to each term in the postings list.

Table IV presents the KWS performance in terms of MTWV of DNN and CNN acoustic models with and without data augmentation on LLPs of Assamese and Haitian Creole. The baseline models are trained without data augmentation. The table is the KWS counterpart of the ASR results in Table I. As can be observed from Table IV, VTLP and SFM not only improve WERs but also MTWVs for the two languages. The improvements are consistent for both DNN and CNN models. Similar to the ASR results, SFM outperforms VTLP for DNN models with MTWV at 0.2826 for Assamese and 0.5021 for Haitian Creole which yields 0.0369 and 0.0495 absolute MTWV improvements for the two language, respectively. On the other hand, VTLP outperforms SFM for CNN models with MTWV at 0.2833 for Assamese and 0.5070 for Haitian Creole which yields 0.0576 and 0.0277 absolute MTWV improvements for the two language, respectively. Furthermore, the two-stage data augmentation combining VTLP and SFM using the stacked architecture achieves the best overall performance with MTWV equal to 0.2862 for Assamese and 0.5178 for Haitian Creole. The results in Tables I and IV have shown the complementariness of VTLP and SFM as data augmentation approaches that help both ASR and KWS.

## V. DISCUSSION

VTLP is representative of a family of data augmentation approaches that are based on altering speech spectra to introduce pattern variations. Typically such approaches are explicitly targeted to some spectral or temporal variations (e.g. VTL or speaking rate) via simple perturbations or distortions. VTLP has a straightforward implementation but is quite effective for dealing with data sparsity as can be observed from the experiments in Section IV. However, DNN/CNN input features may not be very sensitive to small perturbations of VTL. In order for VTLP to make two variations distinctive, the two perturbed VTLs should not be too close to each other. As a consequence, the impact of VTLP can quickly plateau as the number of replicas increases, which has been shown in Table II. For VTLP with random perturbation [18][20], it could result in sampling inefficiency with redundant data samples and prolonged training time.

SFM creates pattern variations in a statistical fashion as opposed to directly altering speech spectra. It assumes that the target speaker's acoustic characteristics are reflected in his/her speaker-dependent acoustic model. SFM is not explicitly targeted to any specific spectral or temporal variation. Instead, the statistically estimated mapping converts the source speaker's "voice" to the target speaker including those acoustic traits that are not trivial to directly perturb or distort. The realization of SFM depends on the input feature space. In this paper we have demonstrated how SFM can be realized in two commonly used input feature spaces–a speaker adaptive feature space for DNN models and LogMEL feature space for CNN models. Ideally, if speaker adaptive training can completely remove the speaker variability in the speaker-adapted feature space, then SFM conducted in this feature space will not be helpful as mapping voice between speakers will not introduce any additional acoustic information from the speaker variability perspective. However, this ideal case is not true practically, which makes SFM helpful in the speaker-adapted feature space as discussed in this paper. For other input features, SFM provides the flexibility of choosing the suitable space in the feature processing pipeline for estimating the speaker dependent model of the target speaker and the mapping function to convert feature sequences.

While VTLP saturates relatively quickly with increased data replicas, SFM plateaus at a slower rate. Gains can still be observed when using 8 replicas in Table II. The quality of SFM apparently relies on the estimation of the speaker dependent model and also the mapping function which, in this paper, are estimated by model space MLLR and constrained MLLR, respectively. These two techniques are among the most widely-used transformation techniques for ASR. In Algorithm I, a generic form of the mapping function is assumed of which CMLLR is only a special form. The same can be said for the model space MLLR with respect to the speaker dependent model. Therefore, we believe that with the improvement of estimation on both the speaker dependent model and the mapping function (e.g. nonlinear transformation), the performance of SFM could be further improved.

The two-stage data augmentation strategy based on the stacked architecture can be used as a general framework to

incrementally combine complementary data augmentation approaches. The second-stage DNN with the speaker adaptive bottleneck feature input is especially suitable for SFM and it can be used together with any potentially complementary data augmentation approach in the first stage to extract bottleneck features. The first stage network in principle can be any network with a bottleneck layer. The reason that the first-stage network is designed as a CNN in this paper is based on the observation that VTLP improves more on CNN models than DNN models.

## VI. Summary

In this paper, we first proposed a label-preserving data augmentation approach based on stochastic feature mapping and investigated it along with vocal tract length perturbation to improve representation of pattern variations in DNN and CNN acoustic modeling to deal with data sparsity. We also proposed a two-stage data augmentation scheme based on a stacked architecture that makes use of the complementariness of the two approaches. Experiments carried out in ASR and KWS on Assamese and Haitian Creole in the context of the IARPA Babel program have shown the effectiveness of the proposed data augmentation approaches.

## Acknowledgment

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

## References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, 2011, pp. 437–440.

[3] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. Autom. Speech Recogn. Understand. Workshop (ASRU)*, 2011, pp. 24–29.

[4] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization," in *Proc. Interspeech*, 2012.

[5] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.

[6] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[7] [Online]. Available: Http://www.iarpa.gov/index.php/research-programs/babel

[8] X. Cui, B. Kingsbury, J. Cui, B. Ramabhadran, A. Rosenberg, M. S. Rasooli, O. Rambow, N. Habash, and V. Goel, "Improving deep neural network acoustic modeling for audio corpus indexing under the IARPA Babel program," in *Proc. Interspeech*, 2013..

[9] M. Karafiat, F. Grezl, M. Hannemann, and J. H. Cernocky, "BUT neural network features for spontaneous Vietnamese in Babel," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 5659–5663.

[10] M. J. F. Gales, K. M. Knill, A. Ragni, and S. P. Rath, "Speech recognition and keyword spotting for low resource languages: Babel project research at CUED," in *Proc. Spoken Lang. Technol. Under-Resource Lang. (SLTU'14)*, 2014..

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998..

[12] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. Int. Conf. Docum. Anal. Recogn. (ICDAR)*, 2003, pp. 958–963.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Neural Inf. Process. Syst. (NIPS)*, pp. 1106–1114, 2012.

[14] R. Lippmann, E. Martin, and D. B. Paul, "Multi-style training for robust isolated-word speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 1987, vol. 12, pp. 705–708.

[15] L. Deng, A. Acero, M. Plumpe, and X. Huang, "Large-vocabulary speech recognition under adverse acoustic environments," in *Proc. Interspeech*, 2000, pp. 806–809.

[16] F.-H. Liu, Y. Gao, L. Gu, and M. Picheny, "Noise robustness in speech to speech translation," in *Proc. Eurospeech*, 2003.

[17] M. J. Hunt and C. Lefebvre, "Distance measures for speech recognition," *Aeronautical Note, NAE-AN-57*, 1989.

[18] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," in *Proc. Int. Conf. Mach. Learn. (ICML) Workshop Deep Learn. Audio, Speech, Lang. Process.*, 2013.

[19] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 5582–5586.

[20] A. Ragni, K. M. Knill, S. P. Rath, and M. J. F. Gales, "Data augmentation for low resource languages," in *Proc. Interspeech*, 2014, pp. 810–814.

[21] Z. Tuske, P. Golik, D. Nolden, R. Schluter, and H. Ney, "Data augmentation, feature combination, and multilingual neural networks to improve ASR and KWS performance for low-resource languages," in *Proc. Interspeech*, 2014, pp. 1420–1424.

[22] N. Kanda, R. Takeda, and Y. Obuchi, "Elastic spectral distortion for low resource speech recognition with deep neural networks," in *Proc. Autom. Speech Recogn. Understand. Workshop (ASRU)*, 2013, pp. 309–314.

[23] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 2, pp. 131–142, Mar. 1998.

[24] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 8, pp. 2222–2235, Nov. 2007.

[25] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 1998, pp. 655–758.

[26] B. Kingsbury, J. Cui, X. Cui, M. J. F. Gales, K. Knill, J. Mamou, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schlüter, A. Sethy, and P. C. Woodland, "A high-performance Cantonese keyword search system," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 8277–8281.

[27] L. Lee and R. Rose, "A frequency warping approach to speaker normalization," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 1, pp. 49–60, Jan. 1998.

[28] M. J. F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 3, pp. 272–281, May 1999.

[29] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput. Speech Lang.*, vol. 12, pp. 75–98, 1998.

[30] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput. Speech Lang.*, vol. 9, pp. 171–185, 1995.

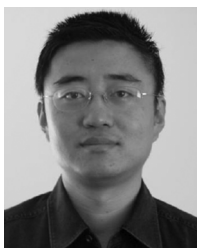
[31] T. N. Sainath, B. Kingsbury, A.-R. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR," in *Proc. Autom. Speech Recogn. Understand. Workshop (ASRU)*, 2013, pp. 315–320.

[32] H. Soltau, G. Saon, and T. N. Sainath, "Joint training of convolutional and non-convolutional neural networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 5572–5576.

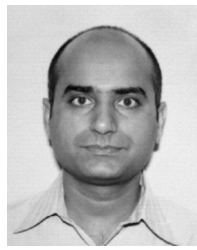
[33] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep convolutional neural network acoustic modeling," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2015, pp. 4545–4549.

[34] T. Yoshioka, A. Ragni, and M. J. F. Gales, "Investigation of unsupervised adaptation for DNN acoustic models with filter bank input," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 6394–6398.

[35] X. Cui, J. Xue, X. Chen, P. A. Olsen, P. L. Dognin, U. V. Chaudhari, J. R. Hershey, and B. Zhou, "Hidden Markov acoustic modeling with bootstrap and restructuring for low-resourced languages," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 8, pp. 2252–2264, Nov. 2012.
[36] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. Spoken Lang. Technol. Workshop (SLT)*, 2010, pp. 97–101.
[37] J. G. Fiscus, J. G. Ajot, J. Garofalo, and G. Doddington, "Results of the 2006 spoken term detection evaluation," in *Proc. SIGIR Workshop Search. Spontan. Conversat. Speech*, 2007, pp. 51–57.
[38] J. Cui, X. Cui, B. Ramabhadran, J. Kim, B. Kingsbury, J. Mamou, L. Mangu, M. Picheny, T. N. Sainath, and A. Sethy, "Developing speech recognition systems for corpus indexing under the IARPA Babel program," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 6753–6757.
[39] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *Proc. Human Lang. Technol. North Amer. Chap. Assoc. Comput. Linguist. (HLT-NAACL)*, 2004, pp. 129–136.
[40] M. Montague and J. A. Aslam, "Relevance score normalization for metasearch," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2001, pp. 427–433.
[41] J. Mamou, J. Cui, X. Cui, M. J. F. Gales, B. Kingsbury, K. Knill, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schlueter, A. Sethy, and P. C. Woodland, "System combination and score normalization for spoken term detection," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 8272–8276.



**Xiaodong Cui** (SM'12) received the B.S. degree (with highest honors) from Shanghai Jiao Tong University, Shanghai, China, in 1996, the M.S. degree from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree from University of California, Los Angeles, in 2005, all in electrical engineering. From 2005 to 2006, he was a Research Staff Member at DSP Solutions R&D Center, Texas Instruments, Dallas, Texas, focusing on noise robust issues for embedded speech recognition systems. Since 2006, he has been a Research Staff Member at Human Language Technologies, IBM T. J. Watson Research Center, Yorktown Heights, NY. His research interests include automatic speech recognition, deep learning in acoustic modeling, keyword search, digital speech processing, machine learning, and pattern recognition.



**Vaibhava Goel** (SM'09) received a B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, and the M.S. and Ph.D. degrees, both in biomedical engineering, from Johns Hopkins University, Baltimore, MD. He has been with the speech group at the IBM T. J. Watson Research Center, Yorktown Heights, NY, since December 2000, where he currently manages research efforts in automatic speech recognition (ASR), audio-visual ASR, and multi-modal deep learning. His primary research interests are in application of statistical modeling and machine learning to speech, vision, and language processing.

**Brian Kingsbury** (SM'09) is a Research Staff Member in the Speech and Language Algorithms department at the IBM T. J. Watson Research Center. He joined IBM Research in 1999 after completing his Ph.D. at the University of California, Berkeley. His research interests include deep neural network acoustic modeling, large-vocabulary speech transcription, and keyword search. He is currently co-PI and technical lead for IBM's efforts in the IARPA Babel program. He has contributed to IBM's entries in numerous competitive evaluations of speech technology, including Switchboard, SPINE, EARS, Spoken Term Detection, and GALE. He is an associate editor for IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING. From 2009 to 2011, he was a member of the Speech and Language Technical Committee of the IEEE Signal Processing Society; he served as a speech area chair for the 2010, 2011, and 2012 ICASSP conferences; and he served as a program chair for the 2014 and 2015 International Conference on Representation Learning (ICLR).